A critical Study on Digital to Analog Conversion and vice versa and introduction to Fuzzy circuit interface

SABYASACHI CHAKRABORTY
Sr. Lecturer and Head of the Department
Department of Computer Science /IT
Icfai University-Sikkim.
Sikkim, India.

Abstract— In this paper first the traditional digital to analog conversion is discussed with circuit analysis. The concepts like propagation delay, inertial delay, step size are discussed. In the later part, the fuzzy logic circuit interface between digital to analog convertor is examined with the help of the latest paper on fuzzy interface. It is shown that the digital to analog conversion process is better carried out if represented by fuzzy logic interface between digital to analog interface.

A new approach to mixed-signal circuit interfacing based on fuzzy logic models is presented. Due to their continuous rather than discrete character, fuzzy logic models offer a significant improvement compared with the classical D-A interface models. Fuzzy logic D-A interfaces can represent the boundary between the digital and analogue worlds accurately without a significant loss of computational efficiency. The potential of mixed-signal interfacing based on fuzzy logic is demonstrated by an example of spike propagation from the digital to analogue world. A model of inertial propagation delay and non-linear DC gain suitable for fuzzy logic gates is also suggested.

Index Terms— Digital Circuit, Analog Circuit, Signal Conversion, Fuzzy Interface, Resolution, Step Size .

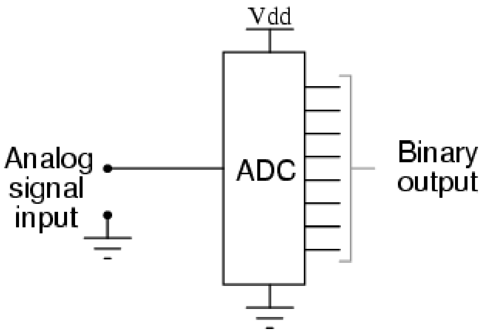
1. INTRODUCTION

The main characteristic of the fuzzy logic is the use of continuous, rather than discrete, waveforms. The advantages of using fuzzy logic for mixed-signal circuit interfacing become evident in situations where the crude propagation delay models do not show spikes
or glitches. Events that generate spikes are sometimes cancelled in classical logic analysis whereas in mixed signal simulations it would be more appropriate to propagate spikes into the analogue part of the circuit. Basic fuzzy logic operations for circuit analysis (such as NOT, AND, OR) can be derived directly from the set operations proposed by Zadeh and others. In the modelling of classical digital-to analogue interfaces it is usually assumed that, for each logic drive to the analogue part of the interface, the corresponding analogue output model switches
between two analogue voltages; one for the driver’s low output state and the other for its high output state.The switching timing is determined by external (logic) signals to the interface’s digital drivers. Models of this type can satisfactorily reflect the effects of digital event cancellations but cannot simulate the
propagation of residual voltage spikes into the analogue domain. The fuzzy-logic digital-to-analogue interface model described here simulates logic events as smooth waveforms and can therefore show small glitches caused by hazards. Continuous fuzzy-logic waveforms lend themselves easily to analogue interfacing and are very well suited for mixed-signal simulations. Fuzzy-logic models have already been applied in behavioural VHDL simulations. Also, system-level behavioural modelling techniques for generic analogue-digital blocks, which have recently been developed , can benefit from the continuous
digital approach offered by fuzzy logic.

2 . ANALYSIS STAGE

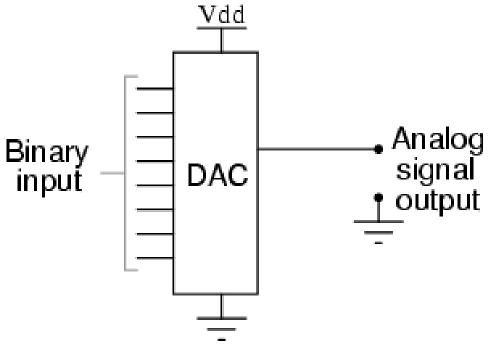
2.1 Analyzing the Conventional System

First of all, let us examine the conventional digital to analog conversion and vice versa. Digital to analog conversion is a process of converting binary bit streams into continuous analog signals such as voltage or current and an analog to digital conversion process inputs an analog electrical signal such as voltage or current and outputs a binary number. In block diagram form, it can be represented as follows:

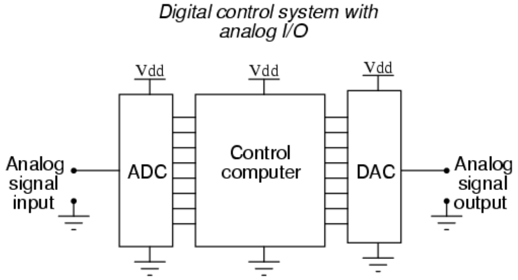


- *Sabyasachi Chakraborty is currently working as Sr.Lecturer(IT/CS) and Head of the Department (IT/CS) in Icfai University-Sikkim-India and currently pursuing Ph.D in Application of Fuzzy Logic in Computer Science and Engineering from West Bengal University of Technology-India Email: scit75@gmail.com,Ph:919635107041. He has 2 Publications in International Journal and 1 in National Conference Proceeding so far. The author has performed and guided over 20 Software Engineerng and Research Projects.*

A DAC, on the other hand, inputs a binary number and outputs an analog voltage or current signal. In block diagram form, it looks like this:



Together, they are often used in digital systems to provide complete interface with analog sensors and output devices for control systems such as those used in automotive engine controls:

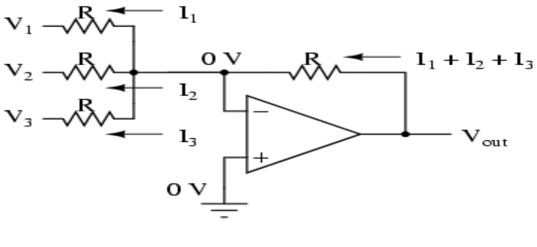


It is much easier to convert a digital signal into an analog signal than it is to do the reverse. Therefore, we will begin with DAC circuitry and then move to ADC circuitry.

The R/2^{n}R DAC

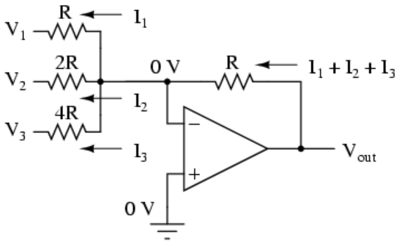
This DAC circuit, otherwise known as the *binary-weighted-input* DAC, is a variation on the inverting summer op-amp circuit. Recalling the classic inverting summer circuit, we find an operational amplifier using negative feedback for controlled gain, with several voltage inputs and one voltage output. The output voltage is the inverted (opposite polarity) sum of all input voltages:

Inverting summer circuit



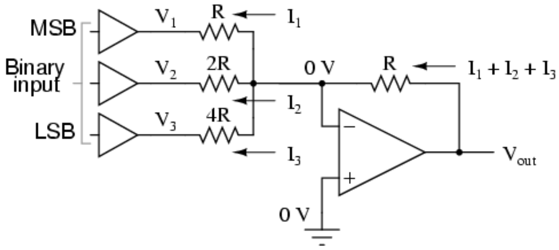
$$V_{out} = -(V_1 + V_2 + V_3)$$

All resistors must be of equal value in case of a simple inverting summer circuit,. If the value of any of the input resistors is different, the input voltages will have different degrees of effect on the output, and the output voltage can not be a true sum.Now let us examine the situation by assuming the different resistor values(R, 2R and 4R)as follows:



$$V_{out} = -(V_1 + \frac{V_2}{2} + \frac{V_3}{4})$$

Starting from V_1 and going through V_3, this would give each input voltage exactly half the effect on the output as the voltage before it. In other words, input voltage V_1 has a 1:1 effect on the output voltage (gain of 1), while input voltage V_2 has half that much effect on the output (a gain of 1/2), and V_3 half of that (a gain of 1/4). These ratios are not arbitrarily chosen: they are the same ratios corresponding to place weights in the binary numeration system. If we drive the inputs of this circuit with digital gates so that each input is either 0 volts or full supply voltage, the output voltage will be an analog representation of the binary value of these three bits.

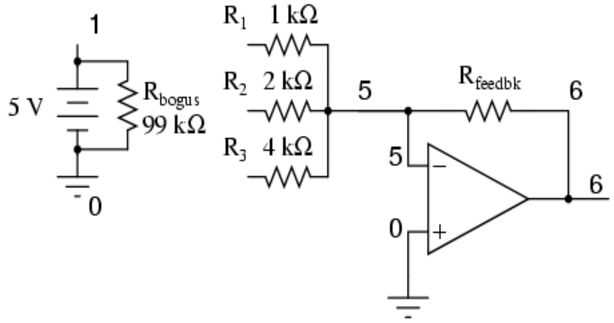


We can list the binary combinations of three inputs and their equivalent output voltages using the above equation as follows:

Binary	Output voltage
000	0.00 V
001	-1.25 V

010	-2.50 V
011	-3.75 V
100	-5.00 V
101	-6.25 V
110	-7.50 V
111	-8.75 V

Note that with each step in the binary count sequence, there results a 1.25 volt change in the output as input varies one unit of decimal value. We can simulate by the DAC circuit with a binary input of 110 where the first node numbers for resistors R_1, R_2, and R_3: a node number of "1" connects it to the positive side of a 5 volt battery, and a node number of "0" connects it to ground . The output voltage appears on node 6 in the simulation:



binary-weighted DAC:

```
v1 1 0 dc 5
rbogus 1 0 99k
r1 1 5 1k
r2 1 5 2k
r3 0 5 4k
rfeedbk 5 6 1k
e1 6 0 5 0 999k
.end
```

```
node  voltage     node  voltage     node  voltage
(1)   5.0000      (5)   0.0000      (6)  -7.5000
```

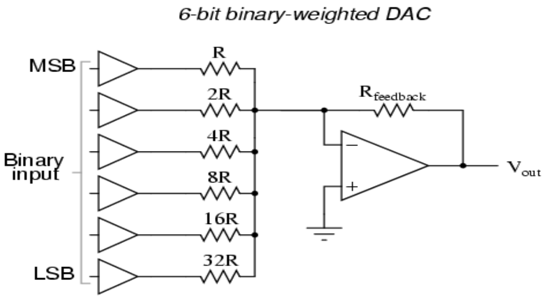
We can adjust resistors values in this circuit to obtain output voltages directly corresponding to the binary input. For example, by making the feedback resistor 800 Ω instead of 1 kΩ, the DAC will output -1 volt for the binary input 001, -4 volts for the binary input 100, -7 volts for the binary input 111, and so on and thus we can say the output volts become 80% of the previous output volts for same input combinations as a result of feedback resistor with 80% of the value of original feedback resistor .

(with feedback resistor set at 800 ohms)

Binary	Output voltage
000	0.00 V
001	-1.00 V
010	-2.00 V
011	-3.00 V
100	-4.00 V
101	-5.00 V

110	-6.00 V
111	-7.00 V

If we wish to expand the resolution of this DAC (add more bits to the input), all we need to do is add more input resistors, holding to the same power-of-two sequence of values:

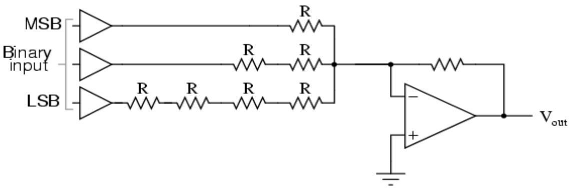


It should be noted that all logic gates must output exactly the same voltages when in the "high" state. If one gate is outputting +5.02 volts for a "high" while another is outputting only +4.86 volts, the analog output of the DAC will be adversely affected. Likewise, all "low" voltage levels should be identical between gates, ideally 0.00 volts exactly. It is recommended that CMOS output gates are used, and that input/feedback resistor values are chosen so as to minimize the amount of current each gate has to source or sink.

The R/2R DAC

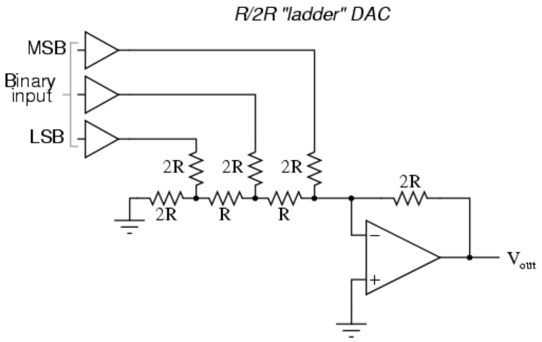
The disadvantage of binary weighted input DAC is the requirement of different values of resistors. The so called R/2R DAC circuit resolves the problem by provision of using lesser variation of resistor values.

We can modify the DAC circuit with resistors R,2R and 4R to use a single input resistance value, by connecting multiple resistors together in series:



Here 2R resistor is replaced by 2 resistors with R value. Clearly we can see , this approach merely substitutes one type of complexity for another: volume of components over diversity of component values. There is, however, a more efficient design methodology.

By constructing a different kind of resistor network on the input of our summing circuit, we can achieve the same kind of binary weighting with only two kinds of resistor values, and with only a modest increase in resistor count. This "ladder" network looks like this:



Mathematically analyzing this ladder network is a bit more complex than for the previous circuit, where each input resistor provided an easily-calculated gain for that bit. For those who are interested in pursuing the intricacies of this circuit further, you may opt to use Thevenin's theorem for each binary input (remember to consider the effects of the *virtual ground*), and/or use a simulation program like SPICE to determine circuit response. Either way, you should obtain the following table of figures:

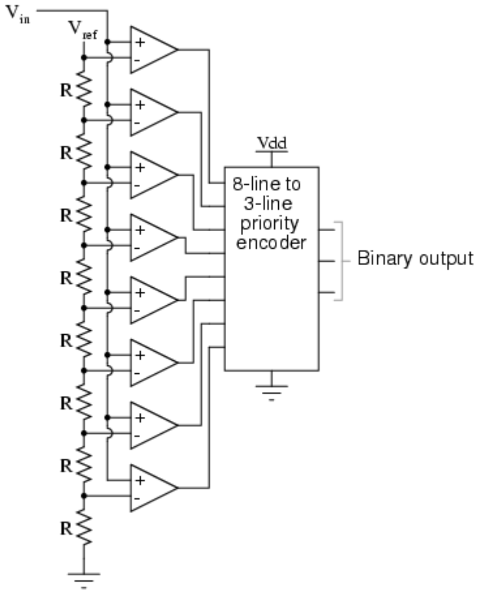
Binary	Output voltage
000	0.00 V
001	-1.25 V
010	-2.50 V
011	-3.75 V

100	-5.00 V
101	-6.25 V
110	-7.50 V
111	-8.75 V

As was the case with the binary-weighted DAC design, we can modify the value of the feedback resistor to obtain any "span" desired. For example, if we're using +5 volts for a "high" voltage level and 0 volts for a "low" voltage level, we can obtain an analog output directly corresponding to the binary input (011 = -3 volts, 101 = -5 volts, 111 = -7 volts, etc.) by using a feedback resistance with a value of 1.6R instead of 2R.

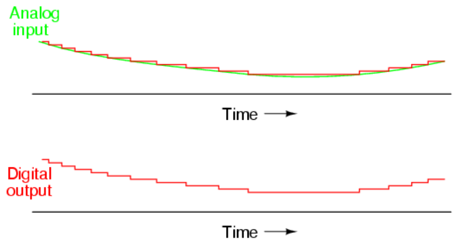
Flash ADC

Also called the *parallel* A/D converter, this circuit is the simplest to understand. It is formed of a series of comparators, each one comparing the input signal to a unique reference voltage. The comparator outputs connect to the inputs of a priority encoder circuit, which then produces a binary output. The following illustration shows a 3-bit flash ADC circuit:

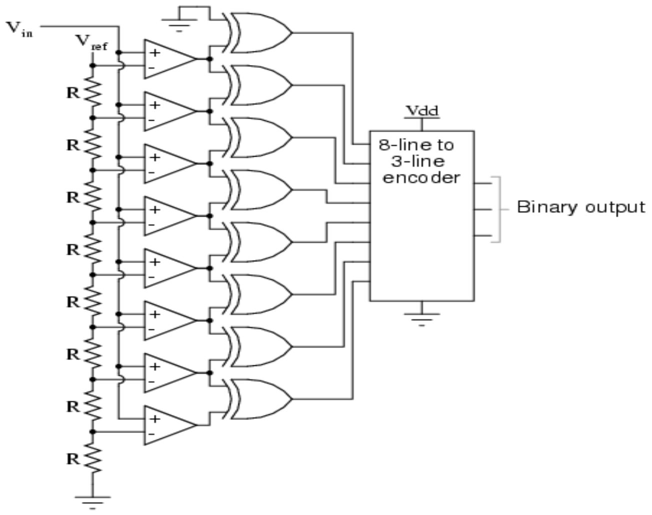


V_{ref} is a stable reference voltage provided by a precision voltage regulator as part of the converter circuit, not shown in the schematic. As the analog input voltage exceeds the reference voltage at each comparator, the comparator outputs will sequentially saturate to a high state. The priority encoder generates a binary number based on the highest-order active input, ignoring all other active inputs.

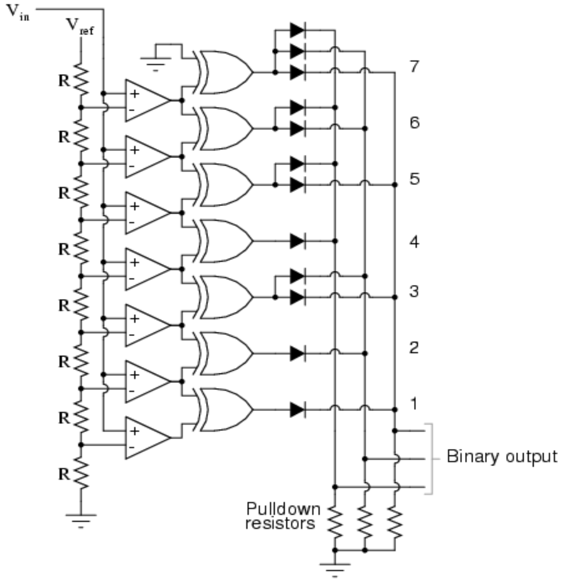
When operated, the flash ADC produces an output that looks something like this:



For this particular application, a regular priority encoder with all its inherent complexity isn't necessary. Due to the nature of the sequential comparator output states (each comparator saturating "high" in sequence from lowest to highest), the same "highest-order-input selection" effect may be realized through a set of Exclusive-OR gates, allowing the use of a simpler, non-priority encoder:



And, of course, the encoder circuit itself can be made from a matrix of diodes, demonstrating just how simply this converter design may be constructed:



Not only is the flash converter the simplest in terms of operational theory, but it is the most efficient of the ADC technologies in terms of speed, being limited only in comparator and gate propagation delays. Unfortunately, it is the most component-intensive for any given number of output bits. This three-bit flash ADC requires eight comparators. A four-bit version would require 16 comparators. With each additional output bit, the number of required comparators doubles. Considering that eight bits is generally considered the minimum necessary for any practical ADC (256 comparators needed!), the flash methodology quickly shows its weakness.

An additional advantage of the flash converter, often overlooked, is the ability for it to produce a non-linear output. With equal-value resistors in the reference voltage divider network, each successive binary count represents the same amount of analog signal increase, providing a proportional response. For special applications, however, the resistor values in the divider network may be made non-equal. This gives the ADC a custom, nonlinear response to the analog input signal. No other ADC design is able to grant this signal-conditioning behavior with just a few component value changes.

2.2 Review of Literature and Critical Analysis with Fuzzy Interface

INTRODUCING FUZZY INTERFACE

Let us consider the list given below :

Binary	Output voltage
000	0.00 V
001	-1.25 V
010	-2.50 V
011	-3.75 V
100	-5.00 V
101	-6.25 V
110	-7.50 V
111	-8.75 V

Here as we have seen earlier that with the change of binary input by 1 decimal value, the absolute magnitude of the voltage changes 1.25 v. If fuzy interface is used then instead of changing 1 decimal value, the input value may change .1 or .2 or .5 decimal value and as a result there will be a very small jump in the output voltage and thus the change in the output voltage within a finite duration will be much more continuous.

The analysis of introducing the fuzzy interface is carried out here with the literature : "**Fuzzy-logic digital-analogue interfaces for accurate mixed-signal simulation" by** Tom Kazmierski, University of Southampton, Southampton, Hampshire, SO17 1BJ, United Kingdom,
In this paper first the traditional digital to analog conversion is discussed with circuit analysis. The concepts like propagation delay, inertial delay, step size are discussed. In the later part, the fuzzy logic circuit interface between digital to analog convertor is examined with the help of the latest paper on fuzzy interface. It is shown that the digital to analog conversion process is better carried out if represented by fuzzy logic interface between digital to analog interface.

In the stated paper a new approach to mixed-signal circuit interfacing based on fuzzy logic models is analyzed. As compared with the traditional Digital-Analog Conversion, The Digial to Analog conversion with fuzzy interface offers a better way as Fuzzy is having continuous rather than the discrete character and thus, it is more suitable to convert a digital signal into its analog form. Fuzzy logic D-A interfaces can represent the boundary between the digital and analogue worlds accurately without a significant loss of computational efficiency as fuzzy can interpret a digital signal properly to produce the equivalent analog signal. In the paper, the example of spike propagation from the digital to analogue world is discussed to demonstrate the potential of mixed-signal interfacing based on fuzzy logic. In this paper a model of inertial delay and the non-linear DC gain is also suggested for fuzzy logic gates.

In the traditional digital to analog and analog to digital conversion process, for general processing of a computer system, The physical input variables are first converted into electrical analog signals using transducers which are represented by continuous wave forms. These electrical analog signals are then transformed into digital signals using Analog to Digital Converter(ADC) . These digital signals are processed by microprocessing unit of the digital computers and the output of the digital computer is converted into analog signals using Digital to Analog Converter(DAC). The output analog signals from DAC are then converted and modified into physical control variables using the actuator.

In basic digital system for digital to analog conversion, the conversion is a process of converting a value represented by a digital data(binary or BCD code) into a voltage or current representing analog signal waves which is proportional to the digital value. For a n bit digital analog converter, there are n input lines carry digital data and thus there are 2^n input combinations. For all these 2^n combinations, there are voltage values to represents them. For example, in case of 3 bit(3 input) DAC, there are 8 input combinations and each combination from 000 to 111 are represented by unique voltages. The analog output voltage is proportional to the digital input combination. Thus Analog output=k . digital input.

Where k is the proportionality factor and is a constant value for a given DAC. The analog output can of course be current or voltage.

Now the output of DAC is not purely analog but a specific value which means it is actually digital which is referred as "pseudo analog" quantity in strict sense. As the number of input bits increases, the number of possible output values will increase and the difference between two output values-which is called as step size will reduce. In this case the produced output is more alike an analog quantity. Thus by increasing the quantity of input bits, the output can be made more like an analog signal. For an example, if a DAC is fed by 4 bit binary counter then the binary counter is continually recycled through its 16 states by applying the clock signal, the output will be like a staircase waveform with a step size of 1V. for 0000 binary input to DAC, the output will be 0V, for 0001 the output will be 1V and likewise for 1111, the output will be 15V which is called as full scale output.

This digital to analog conversion is not perfect as the output will produce a staircase waveform and not exactly continuous analog signal waves. The following paper analyses a better way of digital to analog conversion with the help of fuzzy-analog interface.

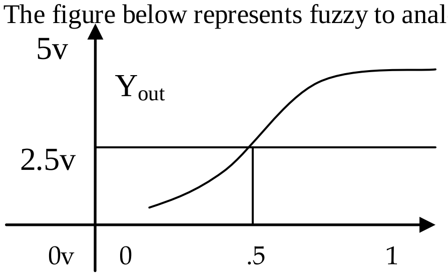
In the research paper titled "**Fuzzy-logic digital-analogue interfaces for accurate mixed-signal simulation"** by Tom Kazmierski,University of Southampton,Southampton, Hampshire, SO17 1BJ,United Kingdom, the significance of fuzzy logic in designing digital to analog and analog to digital interfaces are greatly analysed.

Fuzzy-signal circuit simulation

Circuit simulation based on fuzzy logic is an alternative to the more usual Boolean-logic analysis. Fuzzy operations in logic gates can be defined as :

Fuzzy logic	**Boolean logic**
1 - A	not A
min(A,B)	A and B
max(A,B)	A or B

The figure below represents fuzzy to analog interface.



A sample mapping of fuzzy logic into the analogue domain is shown in Fig. drawn above. This is a representation of Fuzzy XOR gate by analog signals for different voltages between 0v and 5v represented by the intermediate values between 0 and 1.Now there is one very important and unavoidable aspect in fuzzy logis gates that is "Propagation Delays"- which is defined to be the time taken by a pulse of signal to propagate from input terminal to output terminal trough a gate. It is the average transition delay time or the average time of transition of output from low(logic 0) to high(or logic 1) state and from high to low state.

That is to say, Propagation delay time$(t_{pd}) = (t_{pLH} + t_{pHL})/2$.

In this paper, Propagation delays in fuzzy logic gates can be modelled by linear first-order integration and a non-linear DC transfer function.

The inertial delay model which is suggested in the stated paper is nothing but the first-order linear differential equation of the form :

$V_{in} = \tau.\ d (V_d)/dt + V_d$ --------------------------------1.

Where V_{in} is the fuzzy input waveform, V_d is the delayed fuzzy output and □ is the timeconstant.

It represents the sum of total change in delayed output in the given timeline and the initial delayed output.

In a march-in-time simulation with arbitrary input signals V_{in}, the analytic solution of

Eqn. 1 is not generally known and V_d must be found by means of some numerical discretization scheme such as the Backward Euler formula:

$dV(t_{n+1})/dt \approx \{V(t_{n+1})- V(t_n)\}/h$ ---------------------2.

Where h= t_{n+1}- t_n is the simulation step size. The step size here refers to the time gap between n+1 th time and n th time. The equation can be illustrated as the rate of change of voltage at any time n+1 with respect to an unit change in time is equivalent to the change in voltage during the n+1 th period and n th perid in unit step size or with respect to change in n th and n+1 th time. In other words, the rate of change of voltage in n+1 th time point is equivalent to the rate of change of voltage from n th to n+1 th time point.Thus the integrated output signal V_d at the time point t_{n+1} can be calculated as:

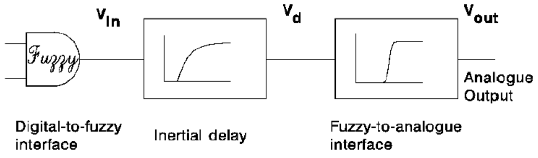
$V_{d,n+1} =\{ h. V_{in,n+1} + \tau. V_{d,n}\}/ (\tau +h)$------------------3.

Where $V_{d,n+1}$ is the new output size and $V_{d,n}$ is the previous output. The equation can be illustrated as the delayed fuzzy output at n+1 th time point is the weighted average of the delayed output of the previous period(n th time point) and the fuzzy input signal of current period(n+1 th time point)

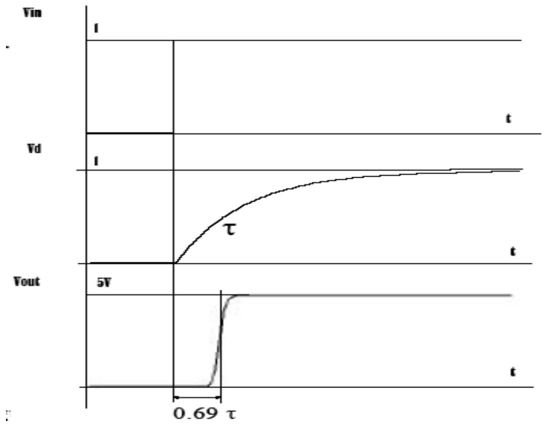
The propagation is a direct result of both the integration and nonlinear transfer characteristic. When the input signal changes its logic value, the change observed on the output is delayed. Assuming that the logic threshold in the non-linear transfer characteristic is equal to 0.5, the propagation delay *d* can be estimated as

$d = \ln 2\, \tau \approx 0.69\, \tau$.

When the integrated waveform $V_d(t)$ exceeds the gate's logic threshold, the output V_{out} (t) begins to change. This model is more accurate than the standard binary-logic model of inertial delay and it corresponds more closely to the performance of real logic gates.



The figure represents Fuzzy to Analog interface with inertial delay.



The figure to represent the propagation delay and the output begins to change as a result of the integrated waveform exceeds the gate's logic threshold.

3 CONCLUSION

Digital to analog conversion and vice versa can be efficiently delivered by using fuzzy interface as revealed in the above study and analysis. There may be further scope of development in regard to fuzzy interfacing can be carried out by significant improvement in inertial delay and propagation delay . The research work undertaken in this paper is basically and majorly a study and survey that relates digital to analog conversion process that converts a binary signal into voltage wave forms and analog to digital conversion to do the reverse to the same using fuzzy interface and how the changes can be noticed. In simple words, as fuzzy variables unlike boolean variables can have the values between 0 and 1 and thus the input values may change in a continuous way and thus the representing voltage will also be continuous rather than discrete. In conventional system, the analog outputs are not perfectly continuous but discrete in nature as there are fixed regular intervels between the voltage values but using fuzzy interface it can be made perfectly continuous.

ACKNOWLEDGMENT

I, as the author want to thank the top management of Icfai University Sikkim for providing with continuous encouragement, support and motivation to carry out the Research Work. I will remain indebted to my PhD guides Dr. Shankar Chandra Ghosh and Dr. Shreeyankar Acharya from West Bengal University of Technology for providing me with enormous support and showing me the proper directions.

REFERENCES

[1] Zadeh, L.A., "Fuzzy sets", Info. Control, no. 8, 1965, 338-353.

[2] Lee R.C.T and C.L. Lang, "Some properties of fuzzy logic", Info. Control, no. 9, 1971, 413-431.

[3] Marinos P.N. "Fuzzy logic and its application to switching systems", IEEE Trans on Computers, C-18, no. 4, 1969, 343-348.

[4] Lee E.T., "Fuzzy logic and the resolution principle", Journal of A.C.M., no 10. 1972, 109-119.

[5] Nichols K.G. , A. D. Brown, M. Zwolinski and T. J. Kazmierski, “Modelling Non-Linear D-A and A-D Interfaces”, Research

- *Sabyasachi Chakraborty is currently working as Sr.Lecturer(IT/CS) and Head of the Department (IT/CS) in Icfai University-Sikkim-India and currently pursuing Ph.D in Application of Fuzzy Logic in Computer Science and Engineering from West Bengal University of Technology-India Email: scit75@gmail.com,Ph:919635107041. He has 2 Publications in International Journal and 1 in National Conference Proceeding so far. The author has performed and guided over 20 software engineerng and Research Projects.*

Journal 1995/96, Department of Electronics and Computer Science, Southampton University, http://www.ecs.soton.ac.uk/rj95/dag/kgn/report.htm.

[6] Zamfirescu A. and C. Ussery, “VHDL and fuzzy-logic if-then rules”, Proc. of Euro-VHDL’92, Hamburg, 1992, 636-641.

[7] Jimenez C.J., A. Barriga, D. Galan, S. Sanchez-Solano, “VHDL package for description of fuzzy-logic controllers”, Proc. of EURO-DAC’95, Brighton, 1995, 528-533.

[8] Genhog G., “A behavioral model of A/D converters using a mixed-mode simulator”, IEEE J. Solid-State Circuits, 26, no. 3, 1991.

[9] Gielen G., E. Liu, A. Sangiovanni-Vincentelli and P. Gray, “Analog behavioral models for simulation and synthesis of mixed-signal systems”, Proc. of EDAC’92, Brussels, 1992, 464-468.